

中華民國經濟部智慧財產局

INTELLECTUAL PROPERTY OFFICE
MINISTRY OF ECONOMIC AFFAIRS
REPUBLIC OF CHINA

茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，
其申請資料如下：

This is to certify that annexed is a true copy from the records of this
office of the application as originally filed which is identified hereunder:

申請日：西元 2003 年 01 月 07 日
Application Date

申請案號：092100252
Application No.

申請人：緯創資通股份有限公司
Applicant(s)

局長
Director General

蔡 練 生

發文日期：西元 2003 年 10 月 13 日
Issue Date

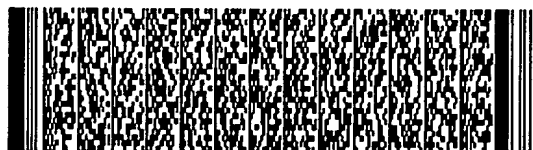
發文字號：09221027960
Serial No.

申請日期：	IPC分類
申請案號：	

(以上各欄由本局填註)

發明專利說明書

一、 發明名稱	中 文	應用程式之核心模式操作系統及方法
	英 文	
二、 發明人 (共1人)	姓 名 (中文)	1. 劉韋宏
	姓 名 (英文)	1. Liu, Wei-Hong
	國 籍 (中英文)	1. 中華民國 TW
	住居所 (中 文)	1. 台北縣汐止市新台五路一段八十八號二十一樓
	住居所 (英 文)	1.
三、 申請人 (共1人)	名稱或 姓 名 (中文)	1. 緯創資通股份有限公司
	名稱或 姓 名 (英文)	1. Wistron Corporation
	國 籍 (中英文)	1. 中華民國 TW
	住居所 (營業所) (中 文)	1. 台北縣汐止市新台五路一段八十八號二十一樓 (本地址與前向貴局申請者相同)
	住居所 (營業所) (英 文)	1.
	代表人 (中文)	1. 林憲銘
	代表人 (英文)	1.



0636_9347TWE(N1);PWHQ_IN_0018_TWXY;Alex Chen_ptd

四、中文發明摘要 (發明名稱：應用程式之核心模式操作系統及方法)

一種應用程式之核心模式操作系統，包括：一核心介面產生器 (Kernel-Mode Interface Generator)、一核心模式介面驅動程式 (Kernel-Mode Interface Driver) 及一特權介面 (Privilege Interface)，核心介面產生器係用以動態產生核心模式介面驅動程式，接著核心模式介面驅動程式產生一呼叫閘道 (Call Gate)，並使呼叫閘道可在一核心模式下執行一最高特權等級 (Ring0) 操作，特權介面係用以連接使用者模式及核心模式，特權介面透過呼叫閘道，提供使用者模式之一行程切換進入核心模式，以執行最高特權等級操作。

伍、(一)、本案代表圖為：第____5____圖

(二)、本案代表圖之元件代表符號簡單說明：

51 ~ 核心介面產生器；

53 ~ 核心模式介面驅動程式；

531 ~ 呼叫閘道；

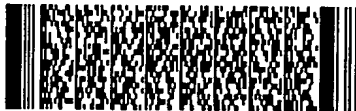
陸、英文發明摘要 (發明名稱：)



四、中文發明摘要 (發明名稱：應用程式之核心模式操作系統及方法)

55 ～ 特 權 介 面 。

陸、英文發明摘要 (發明名稱：)



一、本案已向

國家(地區)申請專利

申請日期

案號

主張專利法第二十四條第一項優先權

二、☐主張專利法第二十五條之一第一項優先權：

申請案號：

日期：

三、主張本案係符合專利法第二十條第一項☐第一款但書或☐第二款但書規定之期間

日期：

四、☐有關微生物已寄存於國外：

寄存國家：

寄存機構：

寄存日期：

寄存號碼：

☐有關微生物已寄存於國內(本局所指定之寄存機構)：

寄存機構：

寄存日期：

寄存號碼：

☐熟習該項技術者易於獲得，不須寄存。



五、發明說明 (1)

【發明所屬之技術領域】

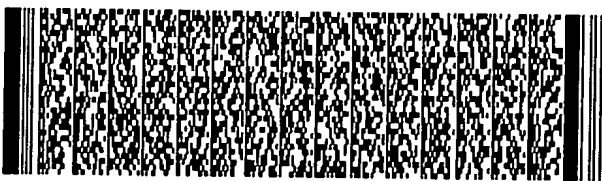
本發明係有關於在保護模式下執行核心模式操作的方法，應用於作業系統中進行低階化的除錯工作，讓使用者易於執行核心模式操作。

【先前技術】

對研發人員來說，常常必須在機器的除錯上花許多時間，投資精力，並且不斷吸收科技新知。就軟體的研發而言，則必須在軟體與硬體間不斷週旋，釐清問題，並解決問題，以求品質的提昇。

在整個產品製造的除錯過程 (debugging process) 中，軟體研發人員所必須扮演的地位，除了軟體本身的問題以外—包含基本輸出／輸入系統 (BIOS)、驅動程式或作業系統，更可能與硬體間有莫大的關聯性。第1圖係顯示磁碟作業系統 (DOS) 環境下之系統資源存取架構圖。DOS之系統資源存取架構包括一應用程式／工具11、一基本輸出輸入系統13、一特權指令15、一輸出入埠 (I/O) 17 以及一記憶體19。在過去的時代，作業系統 (如DOS) 開放了所有的系統資源供使用者任意使用，透過應用程式／工具11直接存取基本輸出輸入系統13或進行低階化的操作 (例如執行特權指令15或對輸出入埠17及記憶體19的存取)，因此軟體研發人員在除錯的過程中，不致於有太大的瓶頸或問題。

然而隨著時代的演進，作業系統為了能夠提昇作業效



五、發明說明 (2)

率，妥善管理系統資源，必須？棄傳統的中央處理器運作模式，即真實模式 (real mode)，而改行32位元的保護模式 (protected mode)。再者，由於作業系統管制了（包含禁止及限制）絕大部份的系統資源，因此對軟體研發人員而言，過去在DOS下所能使用的觀念，在此時則幾乎變成無用武之地，除非軟體研發人員的除錯方式或是工具能夠在保護模式的最高特權等級（稱為Ring0）下執行。為了取得保護模式的最高等級，我們必須知道在保護模式下執行「核心模式操作」（Kernel-Mode operation）的方法。

軟體研發人員為了能夠在現代化的作業系統中進行低階化的除錯工作，或是開發相關軟體，必須認知到現代的作業系統皆是在中央處理器的保護模式下運作的。因此，了解保護模式的基本原理便成為一項基本功夫。且在保護模式的運作下，若欲進行低階化的系統操作，必須讓程式能夠進入到在中央處理器運作之最高特權等級，或是作業系統的核心模式，才有機會使用到全部的系統資源。

第2圖係顯示於視窗作業系統下，藉由核心模式驅動程式執行核心模式操作之架構圖。其中，包括一作業系統之保護模式環境，保護模式環境中包括一應用程式／工具21、一核心模式驅動程式（Kernel-Mode Interface Driver）23、一基本輸出輸入系統251、一特權指令253、一輸出埠257以及一記憶體259。應用程式／工具21及核心模式驅動程式23係於一般使用者模式下執行，基本輸出



五、發明說明 (3)

輸入系統251、特權指令253、輸出入埠257以及記憶體259係存在於作業系統核心模式下。為了能取得最高特權等級，傳統的做法，必須透過應用程式／工具21撰寫核心模式驅動程式23（透過一驅動程式呼叫程序執行核心模式驅動程式23），藉由核心模式驅動程式23執行一系統呼叫（system call）以進入作業系統核心模式，如此便可直接存取基本輸出輸入系統251或進行低階化的操作（例如執行特權指令253或對輸出入埠257及記憶體259的存取）。上述做法必須進入核心模式才能執行這些低階化的系統操作，在微軟的視窗作業系統下，就必須有驅動程式發展工具（DDK，Driver Development Kit）的協助才能完成。

第3圖係顯示透過驅動程式發展工具執行核心模式操作之示意圖。故且不論DDK對驅動程式的分層觀念以及其實作細節，在概念上，它首先必須先將欲在核心模式加以處理的資料予以包裹後，形成輸出入要求封包（IRP，I/O Request Packet）31後，再連同「控制碼」（Control Code）一併以DeviceIoControl()函式的系統呼叫程序通知驅動程式；接著透過輸出／輸入管理系統33，經過層層轉換到達硬體抽象層後35，然後進行硬體37的存取（如特權指令371、輸出入埠373或記憶體375），這樣一個「輸出入要求封包」會經過層層的處理，才會完成整個操作過程。

然而，對於系統製造軟體研發人員而言，這樣繁雜的



五、發明說明 (4)

過程有時會造成困擾，說明如下：(1)大部份的系統製造軟體研發人員所想要的核心模式操作是簡單而直覺的。也許只是幾個輸出入埠的存取，或是必須進入核心模心才能使用的組合語言 (assembly language，如特權指令的執行) 的撰寫；(2)系統研發時期所可能遇到的問題種類複雜，變異性太大，一種為特定需求設計的核心模式驅動程式，不一定能夠應付這些千變萬化的可能性；(3)時效上的爭取對於某些操控項目是必要的，例如一項時間消耗的計算，可能是精準至以毫秒或是一個時脈來計算；而驅動程式的執行路徑若過於冗長，可能影響到時間計算上的精確性；(4)DDK的學習曲線過長，進入門檻值過高，對於只懂C/C++語言或是組合語言的研發人員而言，很難在短時間內了解其撰寫方法；以及(5)原先可在DOS下運作良好的工具，不易輕易運用核心模式驅動程式的方法加以移植，導致研發人員必須為新的軟體架構做大幅度的調整，甚至是重新撰寫原始碼。

基於上述缺點，希望有一個解決策略並可達到以下特性：(a)原始碼可移植性 (source code portability)：原先能在DOS下運作良好的工具，若具有原始碼，最好能夠只作些許的修改，即可在視窗作業系統下運作；(b)語言適用性 (language suitability)：軟體研發人員最好能夠運用即有的C/C++語言，或是組合語言的知識，即可順利在視窗作業系統下立即從事核心模式軟體的撰寫，而不必再學習新的軟體架構；(c)核心操作掌控性



五、發明說明 (5)

(controllability)：如果有必要需使用到非原有知識範圍內的軟體架構，最好能夠輕易掌控其行為；(d)程式可異動性 (variability)：面對千變萬化的除錯需要，最好能夠讓軟體研發人員藉由幾個簡單的調整，即可完成其所要達到的目的；以及(e)操作直覺性 (inductivity)：除了呼應以上各點，同時，最好在視窗作業系統下，一樣能夠以原先在DOS下所擁有的觀念來加以運用。

【發明內容】

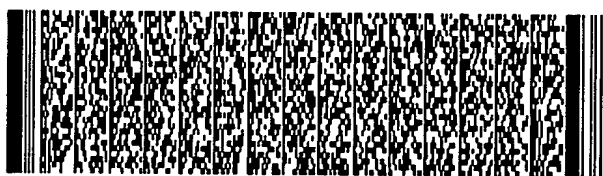
有鑑於此，本發明之目的在提供一種應用程式之核心模式操作系統，使應用程式在使用者模式特權等級 (Ring3) 下執行核心模式操作。

本發明之另一目的在提供一種應用程式之核心模式操作方法，使得軟體發人員能很容易地在保護模式下執行核心模式操作。

基於上述目的，本發明提供一種應用程式之核心模式操作方法，包括下列步驟：提供一核心介面產生器

(Kernel-Mode Interface Generator)；核心介面產生器可產生一核心模式介面驅動程式；核心模式介面驅動程式產生一呼叫閘道，並可使此呼叫閘道具有在核心模式下執行最高特權等級操作之能力；提供一特權介面

(Privilege Interface)，特權介面係用以連接一使用者模式及核心模式；以及藉由呼叫閘道，提供使用者模式之一行程切換進入核心模式，執行最高特權等級之操作。



五、發明說明 (6)

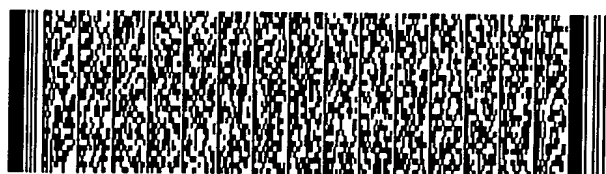
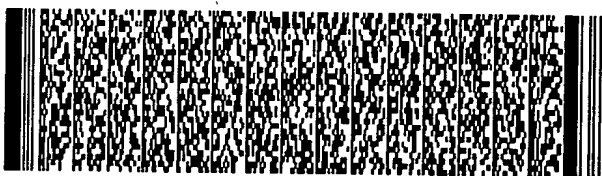
本發明將提供一種能使應用程式在使用者模式特權等級下執行核心模式操作的方法，讓軟體發人員能很容易地在保護模式下執行核心模式操作。

【實施方式】

本發明係提供一種應用程式之核心模式操作系統及方法。

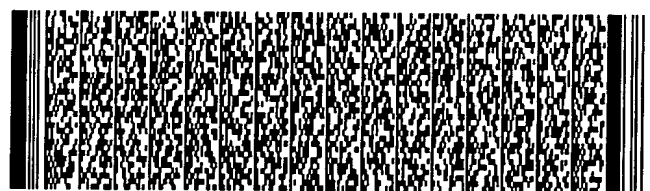
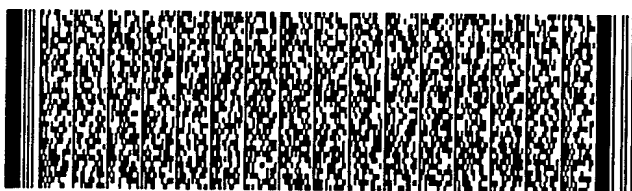
第4圖係顯示作業系統之特權等級之存取權限示意圖。視窗作業系統（如Windows NT/2K/XP）皆為完全的保護模式作業系統，在保護模式中，絕大部份的系統資源不是必須具有足夠的特權（privilege level）才能進行操作，就是被限制使用。這二項限制條件，對於位於使用者模式的行程（User-Mode Process）來說是完全的成立，但是對於核心模式的行程（Kernel-Mode Process）來說，則絕大部份是開放的。意即，一旦進入作業系統的核心模式，等於是取得了中央處理器的最高特權等級，能夠擁有對於絕大部份的系統資源進行操作的權力。因此，為了能夠取得這樣的特權，我們必須找到能夠進入核心模式或是最高特權等級的路徑。

在第4圖中，我們可以知道要取得核心模式的操控權，重點在於能否由使用者模式特權等級的中央處理器特權等級，以合法的路徑進入最高特權等級的特權核心。在英代爾公司（Intel）的IA-32系列的中央處理器的設計中，提供了幾種能夠進入最高特權等級的方法，這些合法



五、發明說明 (7)

的路徑，包括：(i)系統中斷 (system interrupt) 41：工作切換為作業系統在時間間隔 (time-slice) 到達時，所引發的工作本文切換動作 (context switch)。在切換動作發生時，原先的工作中大部份的中央處理器暫存器內容會被保存，並載入下一個工作之中央處理器暫存器內容。在此，我們所提的工作切換為不同特權等級間之工作切換。(ii) 工作切換 (task switch) 43：系統中斷為作業系統或是系統硬體所觸發之工作切換，與前一項工作切換之性質類似，不同的是，此時中央處理器並不負責保存暫存器之內容，而是由中斷服務常式 (Interrupt Service Routine, ISR) 來決定需被保留的暫存器。通常中斷的發生，會伴隨著特權等級的切換，尤其是中斷的性質為硬體中斷 (IRQ)，或是作業系統所引發的中斷或例外 (exception)。(iii) 特權切換指令 (SYSENTER 及 SYSEXIT) 45：此為 Pentium II 以上之 IA-32 中央處理器所提供的一組特別指令，用來使任何的特權等級之工作能夠隨時經由特定的「特權切換指令進入點 (entry point) 451」來取得最高特權等級之權限。通常這組指令必須由作業系統配合做基本的初始化工作，才能正確地被使用。視窗作業系統的設備驅動程式通常使用這組指令來要求作業系統協助完成低階的硬體操作。(iv) 呼叫開道 (Call Gate) 47：此為 IA-32 中央處理器所提供的另一種特權等級切換的機制。這種機制的特色是其可在全域描述器表格 (Global Descriptor Table, GDT) 中配置一組選擇器

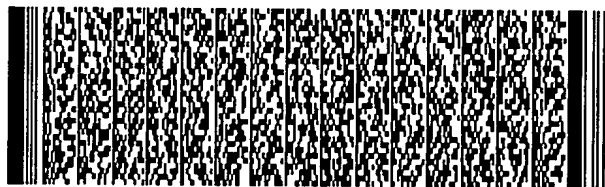


五、發明說明 (8)

(selector) 及呼叫開道進入點471，並開放給符合特權等級要求的工作能夠藉由呼叫開道進入點471來進行等級切換，並在工作完成後，再切換回原先之等級。對執行呼叫的行程來說，呼叫開道的機制就如同一般的程序呼叫般，卻比一般的程序呼叫具有更高的特權。

在以上四種特權等級切換的機制中，對應用程式而言，只有特權切換指令) 45及呼叫開道47兩種方法具有可控制性，亦即可由應用程式決定引發特權切換的時間點。然而，特權切換指令機制的使用，卻必須在使用前先行設定相關的特定模型暫存器 (MSR, Model-Specific Register)，且其僅提供單一進入點的特性（一般來說，作業系統已決定其進入點了），對應用程式而言，不易改變或新增特權進入點。因此，本發明即利用呼叫開道，由應用程式改變或新增特權進入點。

第5圖係顯示本發明之系統架構圖。本發明之系統架構包括一核心介面產生器（亦可稱為KMIf Generator）51、一核心模式介面驅動程式（亦可稱為IKM Driver）53及一特權介面55。核心介面產生器51用來控制產生核心模式介面驅動程式53的生成（construct）及消滅（destruct），並使其與特權介面55產生關聯；核心模式介面驅動程式53由核心介面產生器51動態產生，並用以產生一呼叫開道531及設定呼叫開道531需使用之核心變數的屬性；特權介面55為軟體研發人員所直接接觸的物件類別，為一使用者模式及核心模式間連接的「橋樑」。



五、發明說明 (9)

(bridge)。特權介面55提供了能使在使用者模式運作的行程 (process) 進行核心模式操作的類別方法。

在這些類別方法中，實作了絕大部份會被中央處理器或作業系統認定為「特權的」(privileged)或是「限制的」(restricted)指令，如輸出入埠的讀寫、中斷的致能(enable)及除能(disable)、特定模型暫存器)的讀寫、記憶體的操作…等。這些類別方法皆以「類C」

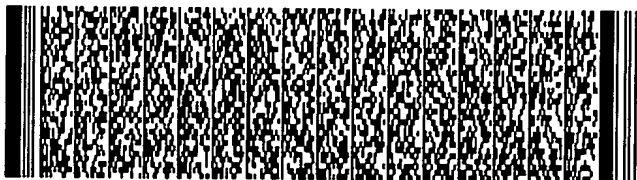
(C-like)的方式來加以命名。因此對於軟體研發人員來說，可以立即上手。另外，對於本發明未直接支援的特權指令，或是某些必須經過特權檢查的系統呼叫(例如，線性位址(linear address)與物理位址(physical address)的相互轉換)，也提供了如Ring0Function()函式之回呼函式(callback function)的間接支援，以應付千變萬化的核心模式操作。

簡言之，本發明係透過特權介面55發出一系統呼叫，藉由呼叫閘道531使特權等級較低的行程取得最高特權等級之權限，以使用特權等級較高的系統資源或操作。

第6圖係顯示呼叫閘道之呼叫流程圖。呼叫閘道在使用上，藉由遠程程序呼叫(call far)指令來完成，如下所述：

```
call far CallGateSelector:CallGateOff
```

其中，CallGateOff並不會被使用到，因此可以為任意值。而CallGateSelector在意義上，係一指向呼叫閘道之遠程指標(far pointer to call gate)，如第6圖所



五、發明說明 (10)

示，其中，呼叫閘道選擇器 (call gate selector) 61 所代表的遠程指標指向全域描述表格 (Global Descriptor Table) 63 之呼叫閘道描述器 (call gate descriptor) 631。在呼叫閘道描述器 631 中，描述呼叫閘道進入點的選擇器資訊，以及可供使用呼叫閘道的特權等級屬性資訊。因此，當呼叫閘道呼叫者 (caller) 執行呼叫時，中央處理器先行判斷呼叫者的特權等級是否足夠，再決定是否取出呼叫閘道所指向的目的程式節區描述器 (code-segment descriptor) 633。如果特權檢查通過了，中央處理器會進行堆疊切換 (stack switch) — 同時隱含著特權等級切換，並將「指令指標」 (instruction pointer) 切換至呼叫閘道所代表的進入點 65 中 (即第 6 圖之程式節區描述器 633 中所存的基底位址 (base address) 加上呼叫閘道描述器 631 中所存的偏移位址 (offset) 後所得的位址)。

一旦呼叫閘道完成指令指標之切換後，同時也完成了特權等級的切換，而核心模式程式介面 (IKM) 對其所配置的呼叫閘道即給予最高特權等級，也因此使得呼叫者擁有最高特權等級的權限。在取得最高的權限後，便能運用任何的系統資源，包含輸出入埠的存取，記憶體存取，特權指令的使用…等等。而在這些核心模式的操作完成後，再以遠程返回 (return far) 來回到呼叫者的返回位址，並回復成原先的特權等級。

本發明提出一個可於視窗作業系統下，以傳統 C/C++ 語言觀念快速撰寫核心模式應用軟體及工具的軟體框架，



五、發明說明 (11)

以解決上述的各種技術困境。

由於本發明的設計理念，是希望能夠讓處於使用者模式特權等級權限的應用程式能完成最高特權等級權限所能做的事，因此，其在呼叫開道的運用上採用輕量原則

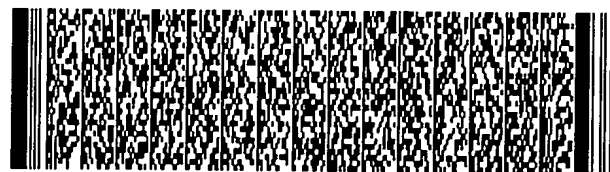
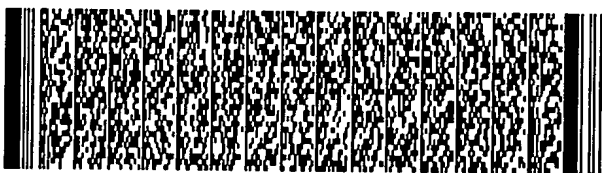
(light-weighted principle)，意即在使用每一個類別方法函式的呼叫時，儘量與中央處理器的指令用途接近，同時將必須在作業系統核心模式才能完成的操作予以實作（如物理位址轉譯為線性位址），以求達到以下效果：

1. 使軟體研發人員能以最直覺的系統操作觀念（即組合語言的觀念，或是輸出入埠或記憶體直接存取的觀念）撰寫核心模式應用程式；

2. 縮短核心模式函式的執行時間，以求爭取最快的時效性（例如：一項操作耗費中央處理器時脈的計算）；

3. 增加核心模式操作的彈性，以求軟體研發人員能夠快速調整核心模式操作的程式細節。

雖然本發明已以較佳實施例揭露如上，然其並非用以限定本發明，任何熟習此技藝者，在不脫離本發明之精神和範圍內，當可作各種之更動與潤飾，因此本發明之保護範圍當視後附之申請專利範圍所界定者為準。



圖式簡單說明

為讓本發明之上述和其他目的、特徵、和優點能更明顯易懂，下文特舉出較佳實施例，並配合所附圖式，作詳細說明如下：

第1圖係顯示磁碟作業系統環境下之系統資源架構圖。

第2圖係顯示於視窗作業系統下，藉由核心模式驅動程式執行核心模式操作之架構圖。

第3圖係顯示透過驅動程式發展工具執行核心模式操作之示意圖。

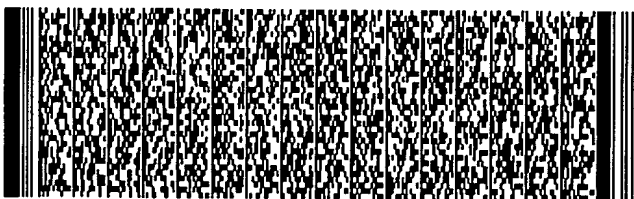
第4圖係顯示作業系統之特權等級之存取權限示意圖。

第5圖係顯示本發明之系統架構圖。

第6圖係顯示呼叫開道之呼叫流程圖。

【符號說明】

- 11、21～應用程式／工具；
- 13、251～基本輸出輸入系統；
- 15、253、371～特權指令；
- 17、257、373～輸出入埠；
- 19、259、375～記憶體；
- 23～核心模式驅動程式；
- 31～輸出／輸入要求封包；
- 33～輸出／輸入管理；



圖式簡單說明

- 35 ~ 硬體抽象層；
- 37 ~ 硬體；
- 41 ~ 系統中斷；
- 43 ~ 工作切換；
- 45 ~ 特權切換指令；
- 451 ~ 特權切換指令進入點；
- 47、531 ~ 呼叫開道；
- 471 ~ 呼叫開道進入點；
- 51 ~ 核心介面產生器；
- 53 ~ 核心模式介面驅動程式；
- 55 ~ 特權介面；
- 61 ~ 遠程指標；
- 63 ~ 全域描述表格；
- 631 ~ 呼叫開道描述器；
- 633 ~ 目的程式節區描述器；
- 65 ~ 進入點。



六、申請專利範圍

1. 一種應用程式之核心模式操作系統，包括：

一核心介面產生器，上述核心介面產生器係用以動態產生一核心模式介面驅動程式，上述核心模式介面驅動程式產生一呼叫開道，並使上述呼叫開道可在一核心模式下執行一最高特權等級操作；以及

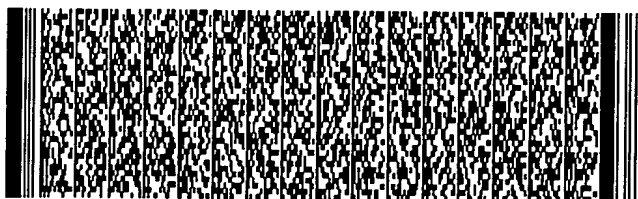
一特權介面，上述特權介面係用以連接一使用者模式及上述核心模式，上述特權介面透過上述呼叫開道，提供上述使用者模式之一行程切換進入上述核心模式，執行上述最高特權等級操作。

2. 如申請專利範圍第1項所述的應用程式之核心模式操作系統，其中，上述特權介面向上述核心介面產生器發出一呼叫開道要求，上述核心介面產生器產生上述核心模式介面驅動程式，接著上述核心模式介面驅動程式產生上述呼叫開道，上述特權介面即透過上述呼叫開道，令上述行程切換進入上述核心模式。

3. 如申請專利範圍第1項所述的應用程式之核心模式操作系統，其中，上述核心模式係位於一中央處理器之一保護模式下。

4. 如申請專利範圍第3項所述的應用程式之核心模式操作系統，其中，上述行程係指一使用者特權等級之操作。

5. 如申請專利範圍第4項所述的應用程式之核心模式操作系統，其中，上述呼叫開道於一全域描述器表格中，配置一選擇器及一特權進入點，以使上述行程於上述核心



六、申請專利範圍

模式中執行上述最高特權等級之操作。

6. 如申請專利範圍第5項所述的應用程式之核心模式操作系統，其中，利用上述全域描述器表格，透過上述選擇器，並經由上述特權進入點，將上述行程之上述使用者特權等級切換至上述最高特權等級，於執行完上述最高特權等級之操作後，切換回上述使用者特權等級。

7. 如申請專利範圍第6項所述的應用程式之核心模式操作系統，其中，上述全域描述器表格有一呼叫開道描述器及一節區描述器。

8. 如申請專利範圍第7項所述的應用程式之核心模式操作系統，其中，上述選擇器所代表之一遠程指標指向上述呼叫開道描述器，當上述呼叫開道之一呼叫者執行呼叫時，上述中央處理器判斷上述呼叫者若具備上述最高特權等級，則將一指令指標切換至上述進入點。

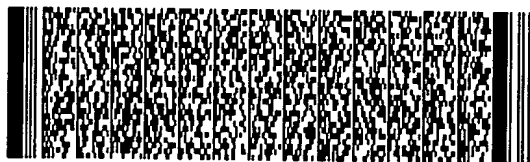
9. 如申請專利範圍第8項所述的應用程式之核心模式操作系統，其中，上述指令指標切換至上述進入點，表示具有上述最高特權等級，可在上述核心模式下執行上述最高特權等級之操作，並在操作完後恢復上述使用者特權等級。

10. 一種應用程式之核心模式操作方法，包括下列步驟：

提供一核心介面產生器；

上述核心介面產生器產生一核心模式介面驅動程式；

上述核心模式介面驅動程式產生一呼叫開道，並使上



六、申請專利範圍

述呼叫閘道可在一核心模式下執行最高特權等級操作；

提供一特權介面，上述特權介面係用以連接一使用者模式及上述核心模式；以及

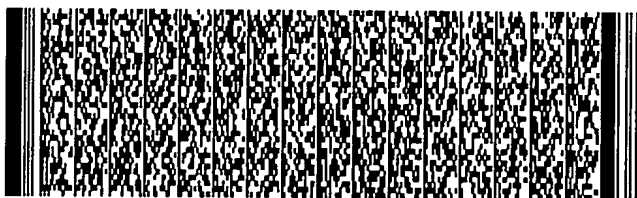
藉由上述呼叫閘道，提供上述使用者模式之一行程切換進入上述核心模式介面驅動程式，執行上述最高特權等級之操作。

11. 如申請專利範圍第10項所述的應用程式之核心模式操作方法，其中，提供上述特權介面以連接上述使用者模式及上述核心模式之步驟中，上述特權介面向上述核心介面產生器發出一呼叫閘道要求，上述核心介面產生器產生上述核心模式介面驅動程式，接著上述核心模式介面驅動程式產生上述呼叫閘道，上述特權介面即透過上述呼叫閘道，令上述行程切換進入上述核心模式。

12. 如申請專利範圍第10項所述的應用程式之核心模式操作方法，其中，上述核心模式係位於一中央處理器之一保護模式下。

13. 如申請專利範圍第12項所述的應用程式之核心模式操作方法，其中，提供上述使用者模式之上述行程切換進入上述核心模式介面驅動程式之步驟中，上述行程係指一使用者特權等級之操作。

14. 如申請專利範圍第13項所述的應用程式之核心模式操作方法，其中，上述呼叫閘道於一全域描述器表格中，配置一選擇器及一特權進入點，以使上述行程於上述核心模式執行上述最高特權等級之操作。



六、申請專利範圍

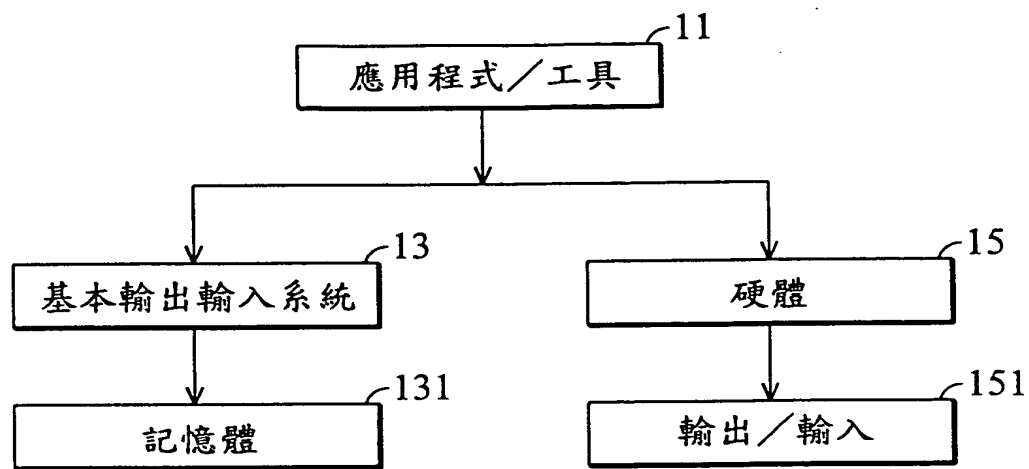
15. 如申請專利範圍第14項所述的應用程式之核心模式操作方法，其中，利用上述全域描述器表格，透過上述選擇器，並經由上述特權進入點，將上述行程之上述使用者特權等級切換至上述最高特權等級，於執行完上述最高特權等級之操作後，切換回上述使用者特權等級。

16. 如申請專利範圍第15項所述的應用程式之核心模式操作方法，其中，上述全域描述器表格有一呼叫閘道描述器及一節區描述器。

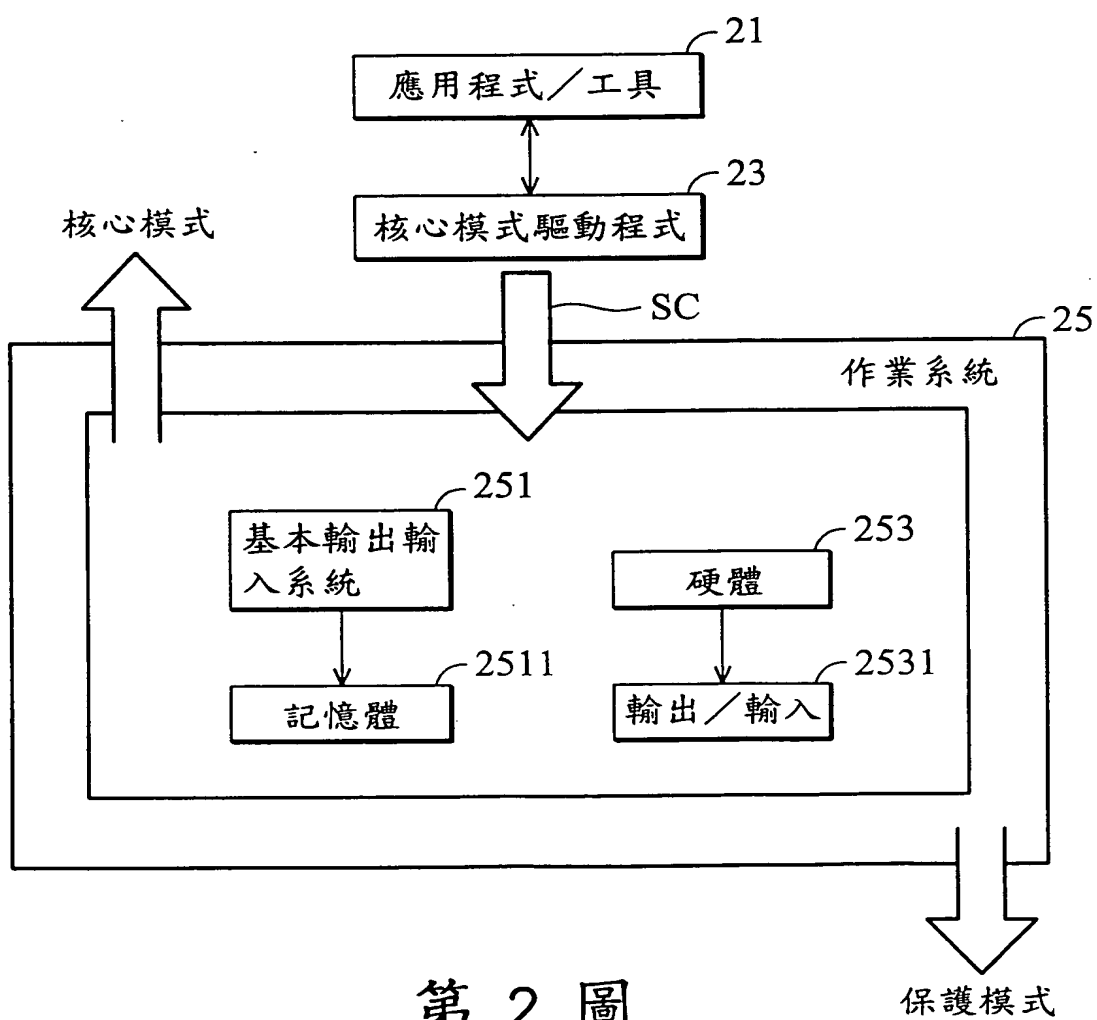
17. 如申請專利範圍第16項所述的應用程式之核心模式操作方法，其中，上述選擇器所代表之一遠程指標指向上述呼叫閘道描述器，當上述呼叫閘道之一呼叫者執行呼叫時，上述中央處理器判斷上述呼叫者若具備上述最高特權等級，則將一指令指標切換至上述進入點。

18. 如申請專利範圍第17項所述的應用程式之核心模式操作方法，其中，上述指令指標切換至上述進入點，表示具有上述最高特權等級，可在上述核心模式下執行上述最高特權等級之操作，並在操作完後恢復上述使用者特權等級。

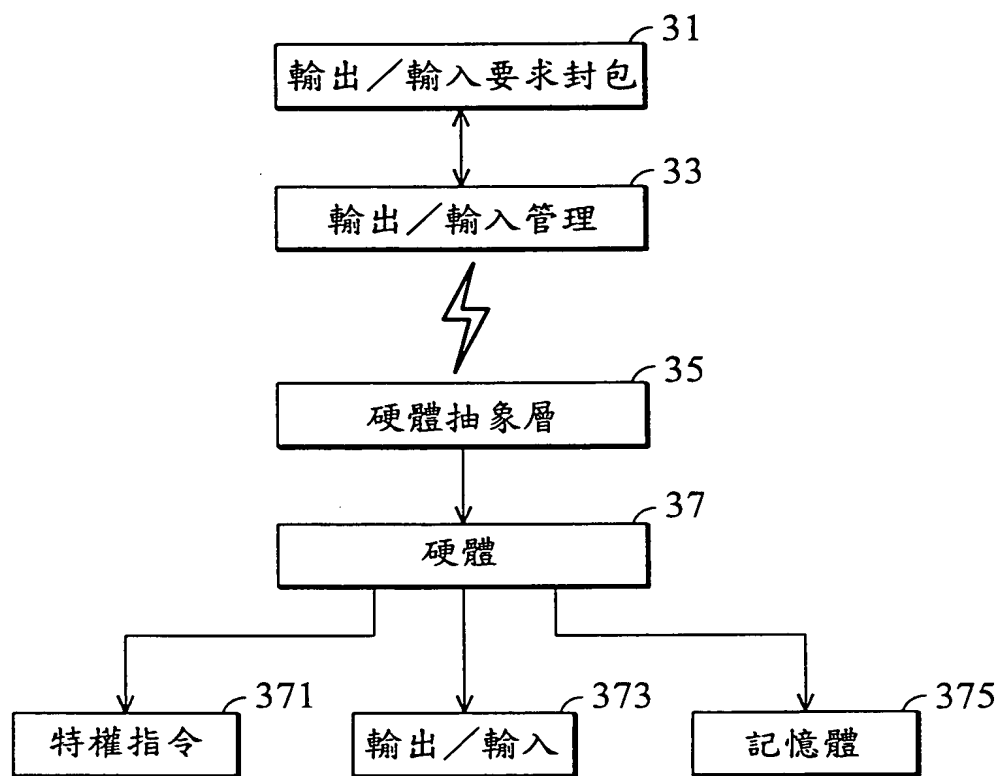




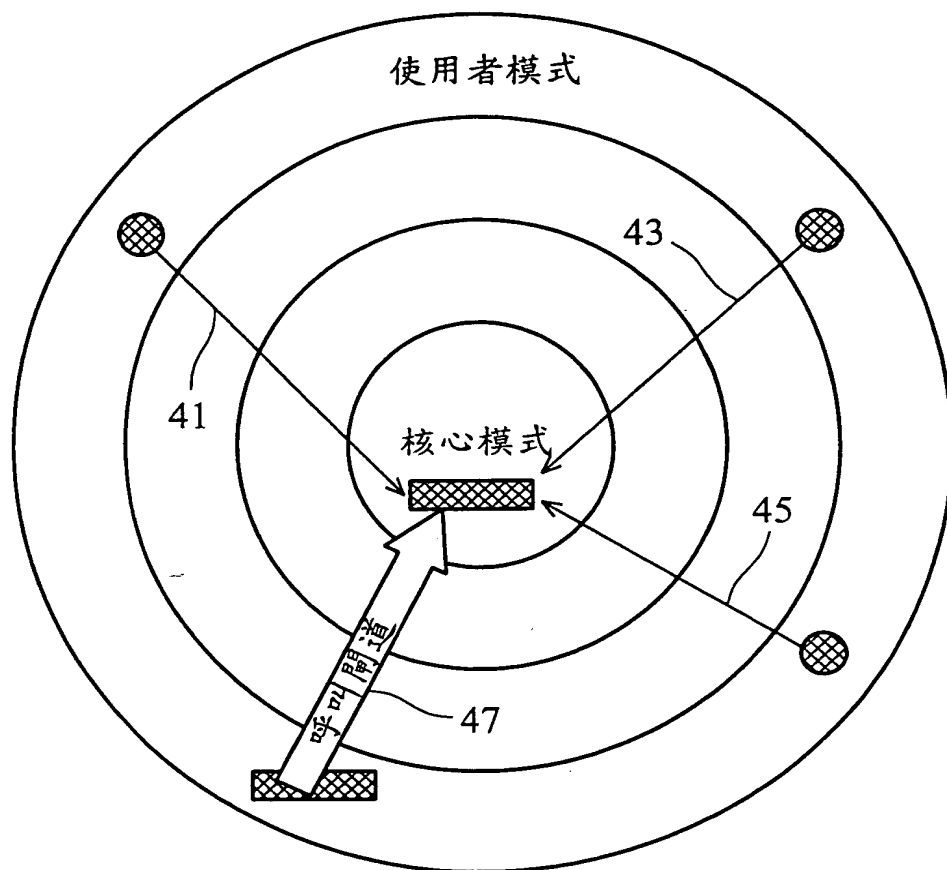
第 1 圖



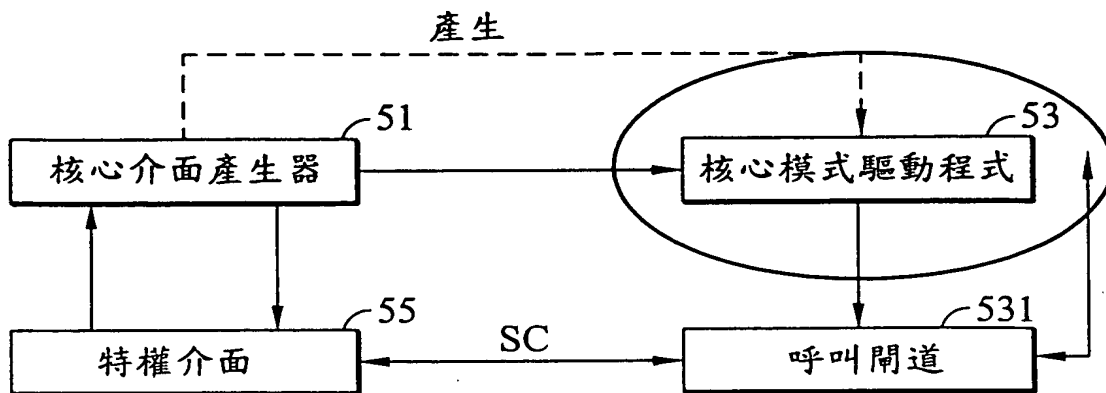
第 2 圖



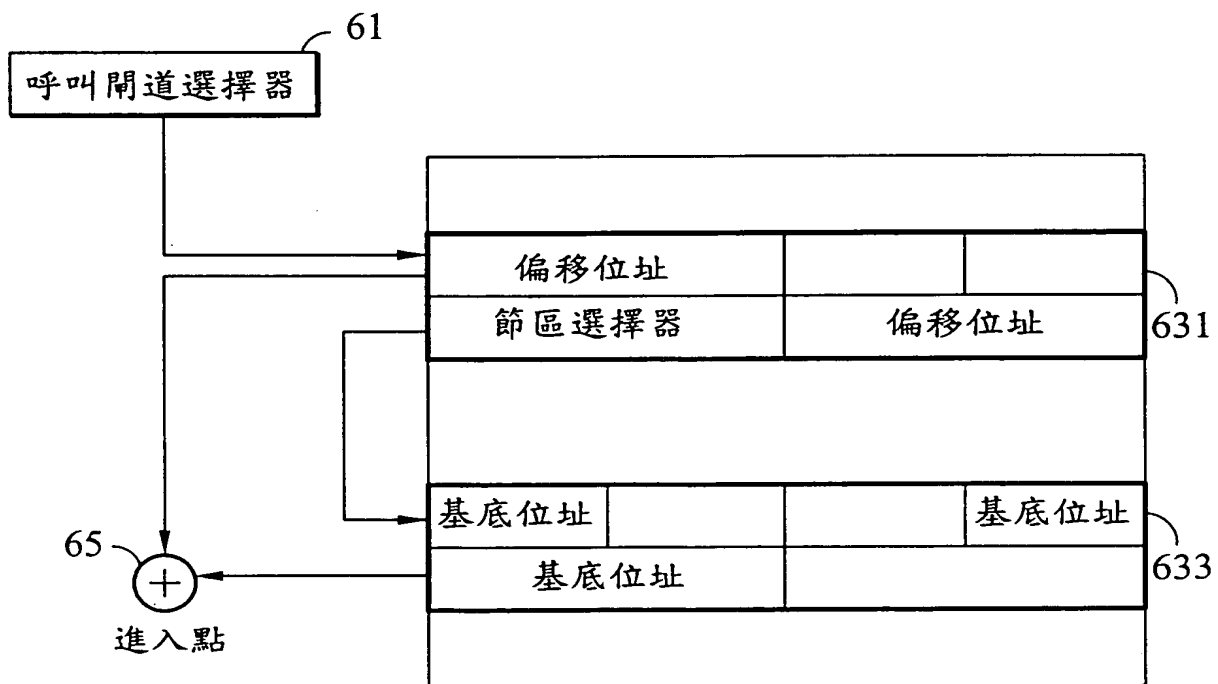
第 3 圖



第 4 圖



第 5 圖

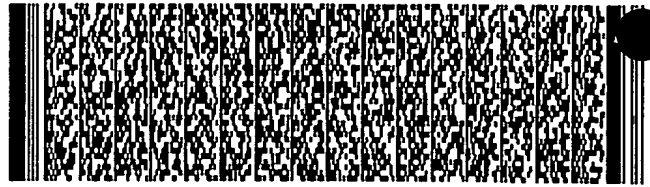


第 6 圖

第 1/21 頁



第 2/21 頁



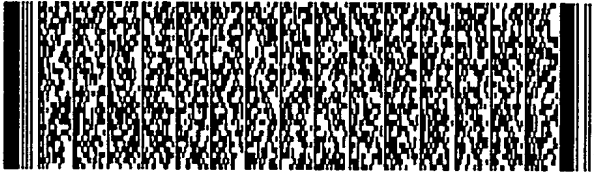
第 3/21 頁



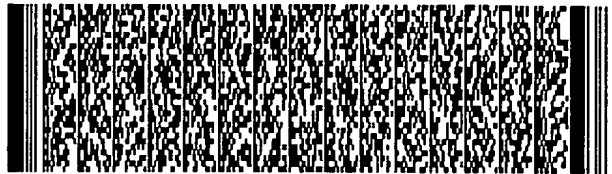
第 4/21 頁



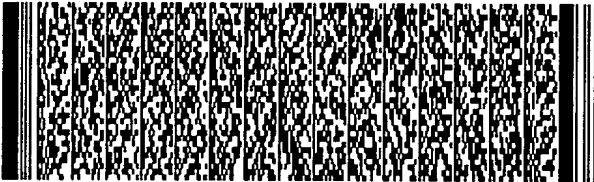
第 5/21 頁



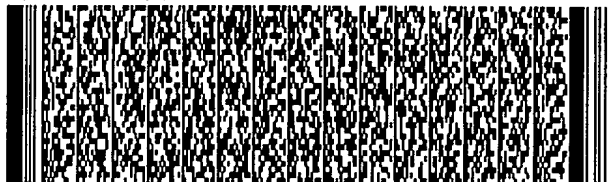
第 5/21 頁



第 6/21 頁



第 6/21 頁



第 7/21 頁



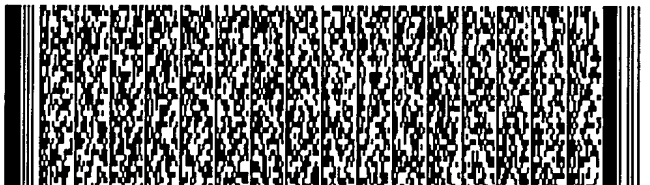
第 7/21 頁



第 8/21 頁



第 8/21 頁



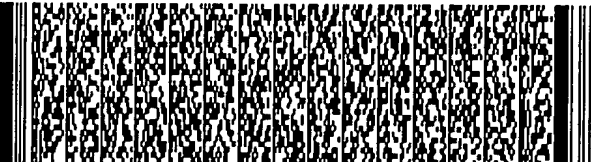
第 9/21 頁



第 9/21 頁



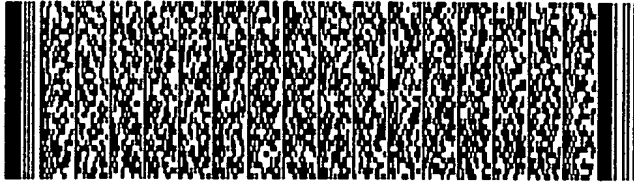
第 10/21 頁



第 10/21 頁



第 11/21 頁



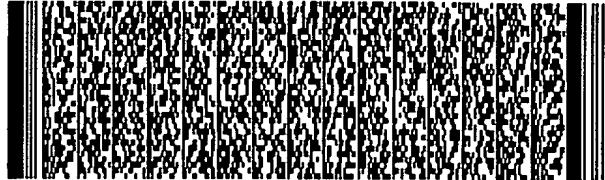
第 11/21 頁



第 12/21 頁



第 12/21 頁



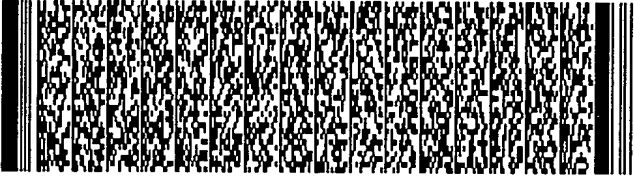
第 13/21 頁



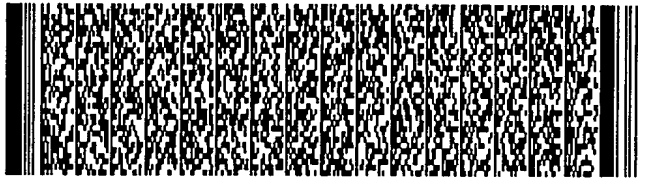
第 13/21 頁



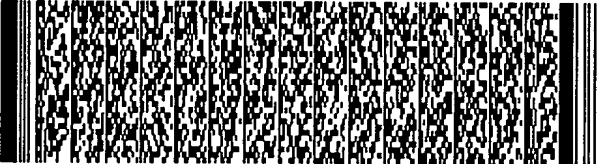
第 14/21 頁



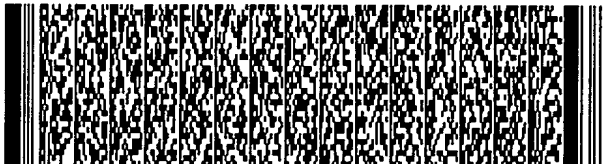
第 14/21 頁



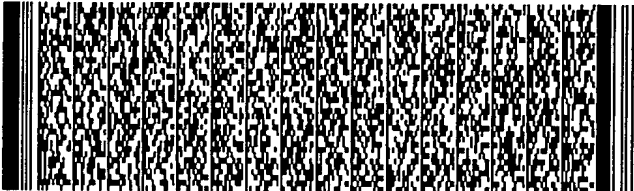
第 15/21 頁



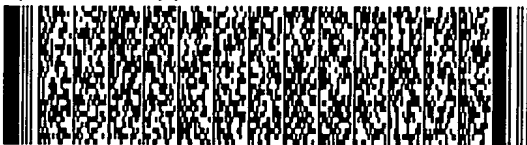
第 15/21 頁



第 16/21 頁



第 17/21 頁



第 18/21 頁



第 19/21 頁



第 19/21 頁



第 20/21 頁

